To set up a direct Ethernet connection between your Raspberry Pi and your Windows 11 PC (step 1, option 1), follow these detailed steps:

### Prerequisites:

- \*\*Raspberry Pi 3B\*\*

- \*\*Ethernet cable\*\* (connects Raspberry Pi to your PC via an Ethernet port or USB Ethernet adapter)

- \*\*USB Hub\*\* (if you need more USB ports for power or peripherals)

- \*\*Power supply for Raspberry Pi\*\* (you can power it via USB from the hub or separately)

### Overview:

We will configure your Raspberry Pi and Windows PC to communicate over Ethernet by:

1. Assigning static IP addresses to both devices.

2. Enabling SSH on the Raspberry Pi to allow remote access.

3. Verifying the connection between the two devices.

### Step-by-Step Guide:

#### 1. \*\*Prepare the Raspberry Pi\*\*

Before connecting the Raspberry Pi to your PC via Ethernet, you need to set up the Pi to allow network access (if this hasn't already been done). There are two ways to do this, depending on whether you already have a display and keyboard attached to your Pi or not:

##### Option 1: If You Have Access to a Display and Keyboard on the Pi:

- \*\*Connect your Pi\*\* to a monitor, keyboard, and mouse.

- \*\*Enable SSH\*\* (SSH allows remote terminal access over the network):

1. Open a terminal window on the Pi.

2. Run the following command to enable SSH:

```bash

sudo raspi-config

```

3. Navigate to "Interfacing Options" > "SSH" > Select "Yes" to enable SSH.

4. Reboot the Raspberry Pi:

```bash

sudo reboot

```

##### Option 2: If You Don’t Have Access to a Display/Keyboard (Headless Setup):

- If you don’t have access to a display, you can enable SSH headlessly by:

1. \*\*Insert the Raspberry Pi's SD card\*\* into your computer.

2. Open the boot partition of the SD card (this is a small partition accessible from Windows).

3. Create a new file named `ssh` (no extension) in the root of the boot partition. This enables SSH automatically when the Pi boots.

#### 2. \*\*Connect the Raspberry Pi to the PC via Ethernet\*\*

- \*\*Connect your Pi to your PC\*\* using an Ethernet cable (through the USB Ethernet adapter on your PC if needed).

- \*\*Power the Raspberry Pi\*\* (via a USB power supply or through the USB hub).

#### 3. \*\*Configure Static IP Addresses\*\*

To make communication between the Pi and your PC easy, we will assign static IP addresses to both devices.

##### A. Configure Windows 11 PC Ethernet Settings:

1. \*\*Open Network Connections\*\*:

- Press `Windows Key + X` and select \*\*Network Connections\*\*.

- In the \*\*Advanced network settings\*\*, click \*\*More network adapter options\*\*.

- Find the \*\*Ethernet adapter\*\* (connected to the Raspberry Pi) and right-click it, then choose \*\*Properties\*\*.

2. \*\*Assign Static IP to Windows 11\*\*:

- In the \*\*Ethernet Properties\*\* window, select \*\*Internet Protocol Version 4 (TCP/IPv4)\*\* and click \*\*Properties\*\*.

- Set the following static IP address:

- \*\*IP address\*\*: `192.168.1.1`

- \*\*Subnet mask\*\*: `255.255.255.0`

- \*\*Default gateway\*\*: Leave blank.

- Click \*\*OK\*\* to save changes.

##### B. Configure Raspberry Pi Static IP:

1. \*\*Access the Pi’s SD card\*\* (if headless, or if using a monitor and keyboard, use the terminal directly).

If using a display, you can edit the network settings directly on the Pi.

Otherwise, insert the SD card into your computer and find the `cmdline.txt` file in the \*\*boot\*\* partition. Here’s how to proceed:

2. \*\*Edit the Network Configuration\*\*:

- In the terminal (if you are working on the Pi), edit the `dhcpcd.conf` file:

```bash

sudo nano /etc/dhcpcd.conf

```

- At the end of the file, add the following lines to configure a static IP for the Pi's Ethernet interface (`eth0`):

```bash

interface eth0

static ip\_address=192.168.1.2/24

static routers=192.168.1.1

static domain\_name\_servers=192.168.1.1

```

3. \*\*Save the file\*\* and reboot the Raspberry Pi:

```bash

sudo reboot

```

Now, your Raspberry Pi should have the IP address `192.168.1.2`, and your PC should be at `192.168.1.1`.

#### 4. \*\*Test the Connection\*\*

After both devices have static IPs, test the connection by pinging the Raspberry Pi from your Windows 11 PC:

1. \*\*Open Command Prompt\*\* on your Windows 11 PC.

2. Type the following command:

```bash

ping 192.168.1.2

```

If the connection is successful, you should see replies from the Raspberry Pi's IP address.

#### 5. \*\*SSH into the Raspberry Pi\*\*

Once the Pi is reachable via ping, you can SSH into it from the Windows 11 PC:

1. Open a terminal on your PC (Command Prompt, PowerShell, or a terminal emulator like \*\*PuTTY\*\*).

2. \*\*Use the SSH command\*\* to connect:

```bash

ssh pi@192.168.1.2

```

- The default username for Raspberry Pi is `pi` and the default password is `raspberry` (if unchanged).

- If prompted, accept the connection by typing "yes".

Now, you should have remote access to your Raspberry Pi through the Ethernet connection.

#### 6. \*\*(Optional) Set Up VNC for GUI Access\*\*

If you want graphical access to your Raspberry Pi’s desktop (in addition to the SSH terminal), you can enable \*\*VNC\*\* for remote desktop:

1. \*\*Enable VNC on the Raspberry Pi\*\*:

- Open terminal on the Pi (via SSH or directly on the Pi if you have a display).

- Run:

```bash

sudo raspi-config

```

- Go to \*\*Interfacing Options\*\* > \*\*VNC\*\* > Enable.

2. \*\*Install VNC Viewer\*\* on your PC:

- Download and install \*\*VNC Viewer\*\* from [RealVNC's website](https://www.realvnc.com/en/connect/download/viewer/).

- Open VNC Viewer, and connect to `192.168.1.2`.

- Enter the Pi's username (`pi`) and password (`raspberry` by default).

#### 7. \*\*Test Robot Controls and Video Streaming\*\*

With the direct Ethernet connection established, you can now proceed to implement your robot control and video streaming as per the instructions provided earlier.

### Troubleshooting:

- \*\*No Ping Response\*\*: If you're unable to ping the Pi, check that the Ethernet cable is securely connected and that both devices have the correct static IP addresses.

- \*\*SSH Issues\*\*: If SSH isn’t working, make sure SSH is enabled on the Pi and check firewall settings on your PC.

- \*\*Network Conflicts\*\*: Ensure no other devices are using the `192.168.1.x` range on the PC or the Pi.

Now your Raspberry Pi is directly connected to your Windows 11 PC over Ethernet, and you can send control commands and receive video streams efficiently.

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

scp -r sentry heroczi@192.168.100.2:/home/heroczi/

workflow:

pc-n szerkesztem a "D:\Dipterv2024\Diplomamunka\_2024\documentation\CODE\sentry" mappában levő fileokat

Mikor kész vagyok nyitok 2 cml-t:

1. A "D:\Dipterv2024\Diplomamunka\_2024\documentation\CODE\" mappában, itt a **scp -r sentry** [**heroczi@192.168.100.2:/home/heroczi/**](mailto:heroczi@192.168.100.2:/home/heroczi/)paranccsal átmásolok mindent a raspberry-re
2. SSH-val belépek a sentryre : **ssh** [**heroczi@192.168.100.2**](mailto:heroczi@192.168.100.2) Ezután bele cd-zek a /home/heroczi/ mappába és futtatom bármelyik python programot

**1. Introduction**

* **Background and Motivation**: Introduce the concept of automated sentry guns, their applications (e.g., security, defense), and why this is an important area of study.
* **Problem Statement**: Define the problem your thesis is addressing, i.e., the need for an automated system to detect, track, and aim at targets autonomously.
* **Objectives**: List the main goals of the project, such as designing a mechanical system, developing hardware and software, and implementing computer vision.
* **Scope and Limitations**: Discuss the scope of your project and any constraints you may face (e.g., target environments, types of objects detected).
* **Thesis Structure**: Provide an overview of the chapters in the thesis.

**2. Literature Review**

* **State of the Art**: Review existing systems related to sentry guns, automated defense systems, and computer vision-based targeting. Include commercial and research-based systems.
* **Related Technologies**: Discuss the technologies behind motion tracking, object detection, and mechatronics systems that are relevant to your project.
* **Gaps in the Literature**: Identify what hasn’t been done or what your project aims to improve upon in terms of performance, automation, or efficiency.

**3. System Design and Requirements**

* **System Overview**: Provide a high-level overview of your automated sentry gun, including its main components (mechanical structure, electronics, software, and sensors).
* **Requirements**: Define both functional and non-functional requirements (e.g., accuracy, response time, power consumption, etc.).
* **Design Constraints**: Describe any physical, budgetary, or technical constraints that influenced the design.

**4. Mechanical Design**

* **3D Modeling and Design**: Detail the process of designing the mechanical parts, such as the gun mount, rotating base, and actuators. Include CAD models and drawings.
* **Material Selection**: Justify the materials used in 3D printing and other mechanical components (e.g., plastic, metal).
* **Kinematics**: Describe the motion control system (e.g., degrees of freedom, rotational and tilting movements).
* **Fabrication and Assembly**: Explain the fabrication process, including 3D printing, assembly of parts, and any challenges encountered.

**5. Hardware Design**

* **Electronic Components**: List and describe the components used (e.g., microcontroller, motors, servos, sensors, power supply).
* **Circuit Design and Schematics**: Provide circuit diagrams for the control system, communication between components, and power management.
* **Hardware Integration**: Explain how the mechanical and electrical components are integrated, including sensor placement and motor control.
* **Testing and Calibration**: Discuss the methods used to test and calibrate the hardware (e.g., motor speeds, positioning accuracy).

**6. Software Development**

* **Control Algorithms**: Explain the algorithms used for targeting, movement control, and decision-making (e.g., PID control for motor movements).
* **Computer Vision System**: Detail the implementation of computer vision for target detection, tracking, and classification. Include techniques like image processing, object detection algorithms (e.g., YOLO, OpenCV, Haar cascades).
* **Software Architecture**: Provide an overview of the software design, including modules for vision, motor control, and communication.
* **Programming Languages and Tools**: List the languages (e.g., Python, C++) and development tools (e.g., TensorFlow, OpenCV) used for implementation.

**7. System Integration and Communication**

* **Communication Protocols**: Discuss the communication between the hardware components (e.g., microcontroller and computer vision system), whether it's via UART, I2C, or wireless protocols.
* **Synchronization**: Explain how different components are synchronized to ensure the sentry gun operates efficiently and accurately.
* **Data Flow**: Show the flow of data within the system, from sensors to decision-making and actuation.

**8. Testing and Validation**

* **Test Setup**: Describe the test environment and conditions (e.g., indoor/outdoor, lighting, object distances) used to validate the system’s performance.
* **Performance Metrics**: Define the metrics used to evaluate the system, such as accuracy of target detection, response time, tracking efficiency, and power consumption.
* **Experimental Results**: Present the results of the tests, including performance under different conditions (e.g., varying lighting, multiple targets, moving objects).
* **Analysis of Results**: Analyze the results, highlighting areas where the system performed well and where improvements are needed.

**9. Challenges and Limitations**

* **Challenges Encountered**: Discuss the technical and practical challenges faced during the project, such as hardware failures, software bugs, or limitations of computer vision algorithms.
* **Limitations of the System**: Acknowledge the limitations of the current implementation, such as detection range, environmental constraints (e.g., weather conditions), or the system's response time.

**10. Future Work**

* **Improvements**: Suggest potential improvements in the design, hardware, and software. For example, using higher-resolution cameras, more powerful processors, or advanced computer vision algorithms (e.g., deep learning-based object detection).
* **Additional Features**: Propose new features that could be added, such as enhanced object recognition, more sophisticated decision-making systems (e.g., distinguishing between friend and foe), or improved mobility for the sentry gun.

**11. Conclusion**

* **Summary of Contributions**: Summarize the main contributions of your thesis, such as the design of the automated sentry gun and its novel features.
* **Cost**
* **Final Remarks**: Provide a final reflection on the project, its importance, and potential impact on the field of mechatronics or security systems.

**12. References**

* List all the research papers, books, software libraries, and tools you referenced throughout your thesis. Follow a standardized citation format.

**13. Appendices**

* **Detailed Schematics**: Include full mechanical and electronic schematics that were too detailed for the main body of the text.
* **Code Listings**: Provide key portions of the code that drive the system (e.g., motor control, computer vision algorithms) for reference.
* **Datasheets and Specifications**: Attach datasheets of important components, like motors, sensors, and microcontrollers.